Overview

The LONTALK Protocol is designed to support the needs of applications spanning a range of industries and requirements. To meet its broad objectives, the protocol is presented to programmers and installers as a collection of services that may be optionally invoked. Services may be chosen by the programmer and fixed at compile time. In addition, many of the service choices may be changed by an installer when a node is installed or reconfigured in a particular LONWORKS application.

The LONTALK Protocol follows the reference model for open systems interconnection (OSI) developed by the International Standard Organization (ISO). In the terminology of the ISO the LONTALK Protocol provides services at all 7 layers of the OSI reference model as shown below:

- Physical channel management (layers 1 and 2)
- Naming, addressing, and routing (layers 3 and 6)
- Reliable communications and efficient use of channel bandwidth (layers 2 and 4)
- Priority (layer 2)
- Remote actions (layer 5)
- Authentication (layers 4 and 5)
- Network management (layer 5)
- Foreign Frame transport and data interpretation (layer 6)
- Application Compatibility (layer 7)

The details of these services are described in this chapter. Table 1 summarizes the OSI reference model layers and the LONTALK services provided at each layer.

Table 1. LONTALK Protocol Layering

	OSI Layer	Purpose	Services Provided
7	Application	Application Compatibility	Standard Network Variable Types
6	Presentation	Data Interpretation	Network Variables Foreign Frame Transmission
5	Session	Remote Actions	Request-Response Authentication Network Management
4	Transport	End-to-End Reliability	Acknowledged & Unacknowledged Unicast & Multicast Authentication Common Ordering; Duplicate Detection
3	Network	Destination Addressing	Addressing Routers
2	Link	Media Access and Framing	Framing, Data Encoding; CRC Error Checking Predictive CSMA; Collision Avoidance; Optional Priority & Collision Detection
1	Physical	Electrical Interconnect	Media-Specific Interfaces and Modulation Schemes (twisted pair, powerline, radio frequency, coaxial cable, infrared, fiber optic)

The Physical Channel

The LONTALK protocol supports networks with segments using differing media. The media supported by the LONTALK protocol include twisted pair, powerline, radio frequency, infrared, coaxial cable, and fiber optics. The specifications for each LONWORKS transceiver provides the distance, data rates, and topologies supported.

Every LONWORKS node is physically connected to a channel. A channel is a physical transport medium for packets; a LONWORKS network is composed of one or more channels. The physical form of a channel depends on the medium. For example, a twisted pair channel is a twisted pair wire; an RF channel is a specific radio frequency; a powerline channel is a contiguous section of AC power wiring.

Multiple channels are connected by bridges and routers. Bridges and routers consist of two NEURON CHIPS connected together via their I/O pins. Each of the NEURON CHIPS is connected to a different channel via an appropriate transceiver; a transceiver provides the interface between a channel and a NEURON CHIP.

The data rate of a channel is dependent upon the medium and transceiver design. Multiple transceivers with different data rates may be designed for a medium to allow trade-offs of distance, throughput, and node power consumption and cost.

The transaction throughput supported by a given channel is limited by several factors in addition to the channel data rate. At low data rates or with long packets, the packet transmission time and average media access delay form the bounds of packet throughput. At higher data rates with short packets, the packet processing power of the NEURON CHIP limits channel performance.

Tables 2 and 3 estimate the approximate network throughput as a function of data rate and packet size. The "peak" traffic numbers can be supported for short bursts.

Table 2. LONTALK Protocol Channel Throughput — 12 byte packets

Data Rate (K Bits/Sec)	Peak Number of Packets/Sec	Sustained Number of Packets/Sec
4.883	25	20
9.766	45	35
19.531	110	85
39.063	225	180
78.125	400	320
156.25	625	500
312.5	700	560
625.0	700	560
1,250.0	700	560

Table 3. LONTALK Protocol Channel Throughput — 64 byte packets

Data Rate (K Bits/Sec)	Peak Number of Packets/Sec	Sustained Number of Packets/Sec
4.883	7	5
9.766	13	10
19.531	25	20
39.063	50	40
78.125	100	80
156.25	200	160
312.5	340	270
625.0	500	470
1,250.0	700	560

Naming, Addressing, and Routing

A name is an identifier that uniquely identifies a single object within an object class. A name is assigned when an object is created and does not change over its lifetime. The 48-bit unique ID is a name for a NEURON CHIP because it uniquely distinguishes a NEURON CHIP from all other NEURON CHIPS, and does not change over the lifetime of the NEURON CHIP.

An address is an identifier that uniquely identifies an object or group of objects within an object class. Unlike a name, an address may be assigned and changed any time after an object is created.

LONTALK addresses uniquely identify the source node and destination node (or nodes) of a LONTALK packet. These addresses are also used by routers to selectively pass packets between two channels.

A NEURON CHIP's unique ID may be used as an address. However, the NEURON CHIP's unique ID is not used as the sole form of addressing in the LONTALK protocol because such addressing only supports one-to-one transactions (i.e., no groups), and routing tables based on unique node addresses would have to be very large. This addressing mode is used primarily during installation and configuration, since it allows communications with individual nodes when the network topology is not yet known.

To simplify routing, the LONTALK protocol defines an hierarchical form of addressing using domain, subnet, and node addresses. This form of addressing can be used to address the entire domain, an individual subnet, or an individual node. To further facilitate the addressing of multiple dispersed nodes, the LONTALK protocol defines another class of addresses using domain and group addresses.

This also simplifies replacement of nodes in a functioning network. The replacement node is assigned the same address as the node it replaces. Thus all references to this node from elsewhere on the network do not need to be modified, as would be the case if Unique ID addressing were to be used.

The various address forms are described in the following sections, along with discussions on routers and address generation.

The Domain Address Component

A domain is a logical collection of nodes on one or more channels. Communications can only take place among nodes configured in a common domain; therefore, a domain forms a virtual network. Multiple domains can occupy the same channels, so domains may be used to prevent interference between nodes in different networks.

For example, two adjacent buildings using nodes with RF transceivers on the same frequency would be on the same channel. To prevent interference between the applications carried out by the nodes, the nodes in each building would be configured to belong to different domains.

The NEURON CHIP may be configured so that a LONWORKS node may belong to one or two domains. A node that is a member of two domains may be used as a gateway between the two domains. The LONTALK protocol does not support communications between domains, but an application program may be implemented to forward packets between two domains.

A domain is identified by a domain ID. The domain ID may be configured as 0, 1, 3, or 6 bytes. Six byte domain IDs can be used to ensure that the domain ID is unique: for example, using the 48-bit ID of one of the NEURON CHIPS in the domain as the domain ID ensures that no other network can have the same domain ID, since all NEURON CHIP IDs are unique. However, six byte domain IDs add six bytes of overhead to every packet. The overhead may be reduced by using a shorter domain ID. In a system where there is no possibility of interference between multiple networks, the domain ID may be configured as zero bytes. For example, LONWORKS applications using twisted-pair channels may be configured with zero-byte domains if only one application will be using the twisted pair channels. Domain IDs may be configured as 1 or 3 bytes in systems where a single administrator controls assignment of domain IDs to prevent duplicate IDs.

The domain ID can also be used for application-level purposes. For example, a domain ID could be used by service personnel as a system identifier.

The Subnet Address Component

A subnet is a logical collection of up to 127 nodes within a domain. Up to 255 subnets can be defined within a single domain. All nodes in a subnet must be on the same channel, or on channels connected with bridges. Subnets cannot cross routers.

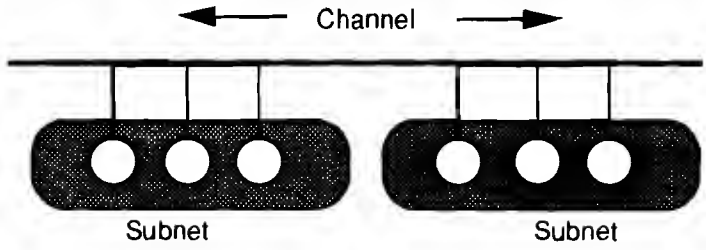


Figure 1. Multiple subnets on a common channel

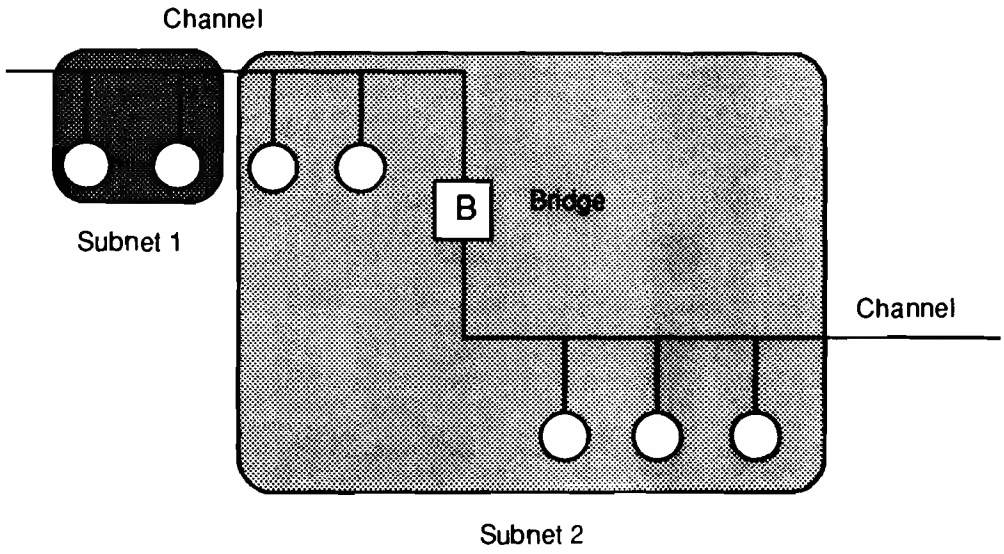


Figure 2. Subnets can span bridges

If a node is configured to belong to two domains, it must be assigned to a subnet within each of the domains.

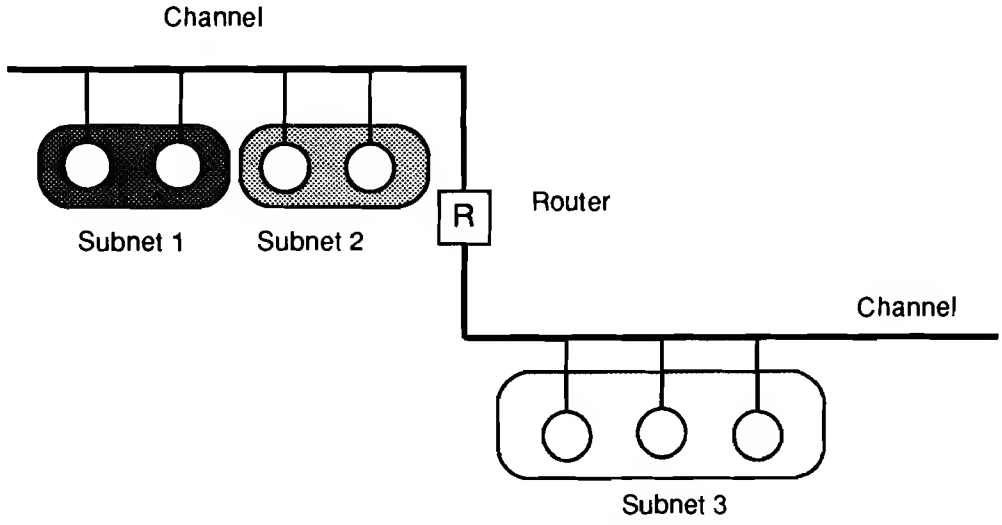


Figure 3. Subnets cannot span routers

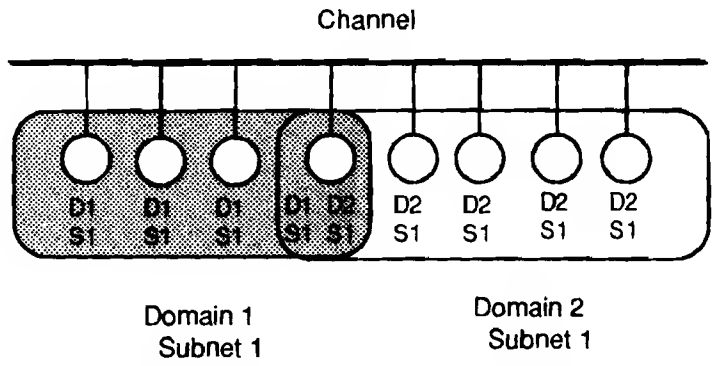


Figure 4. A node configured in 2 domains has subnet assignments for each domain

All nodes within a domain are typically configured in the same subnet except in the following cases:

- They are located on different channels with intervening routers. Since subnets cannot cross routers, the nodes must be on different subnets.
- Configuring the nodes in the same subnet would exceed the maximum number of nodes allowed in a subnet. Subnets are limited to 127 nodes. Multiple subnets may be configured on a set of channels connected by bridges to increase the capacity of the channels above 127 nodes. For example, a set of channels connected by bridges with two subnets may have up to 254 nodes; three subnets may have up to 381 nodes.

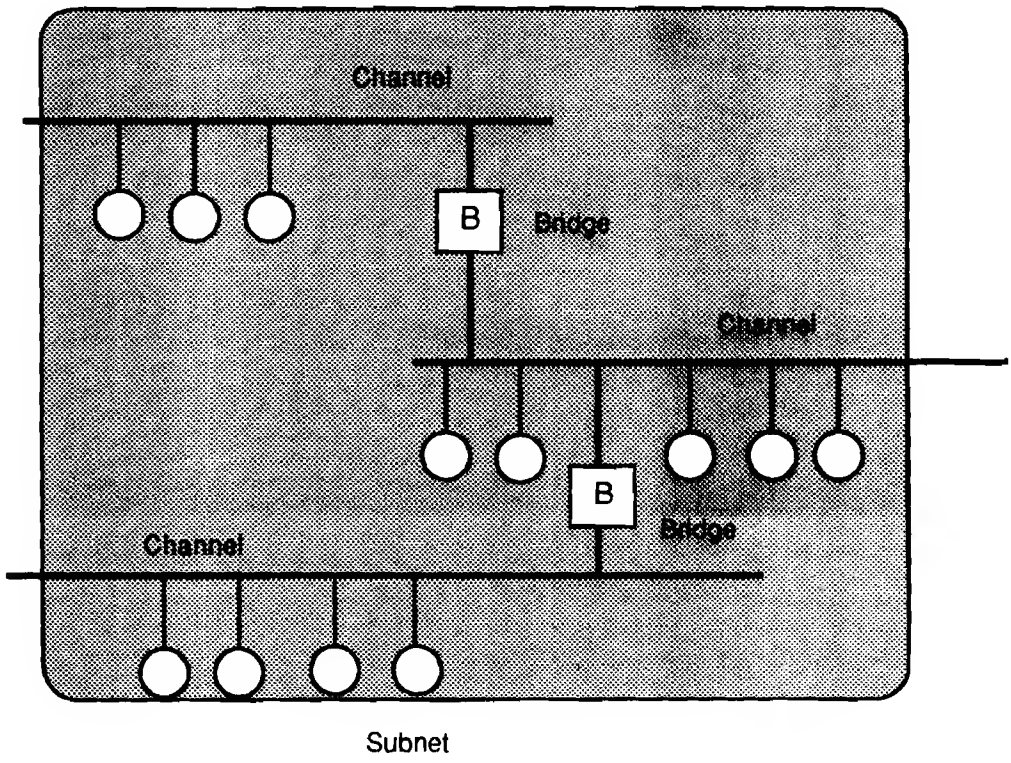


Figure 5. All nodes in one domain configured as a single subnet

The Node Address Component

Every node within a subnet is assigned a unique node number within that subnet. The node number is 7 bits, so there may be up to 127 nodes per subnet. A maximum of 32,385 nodes (255 subnets x 127 nodes per subnet) may be in a single domain.

Groups

A group is a logical collection of nodes within a domain. Unlike a subnet, however, nodes are grouped together without regard for their physical location in the domain. The NEURON CHIP allows a node to be configured to be a member of up to 15 groups.

Groups are an efficient way to use network bandwidth for one-to-many network variable and message tag connections.

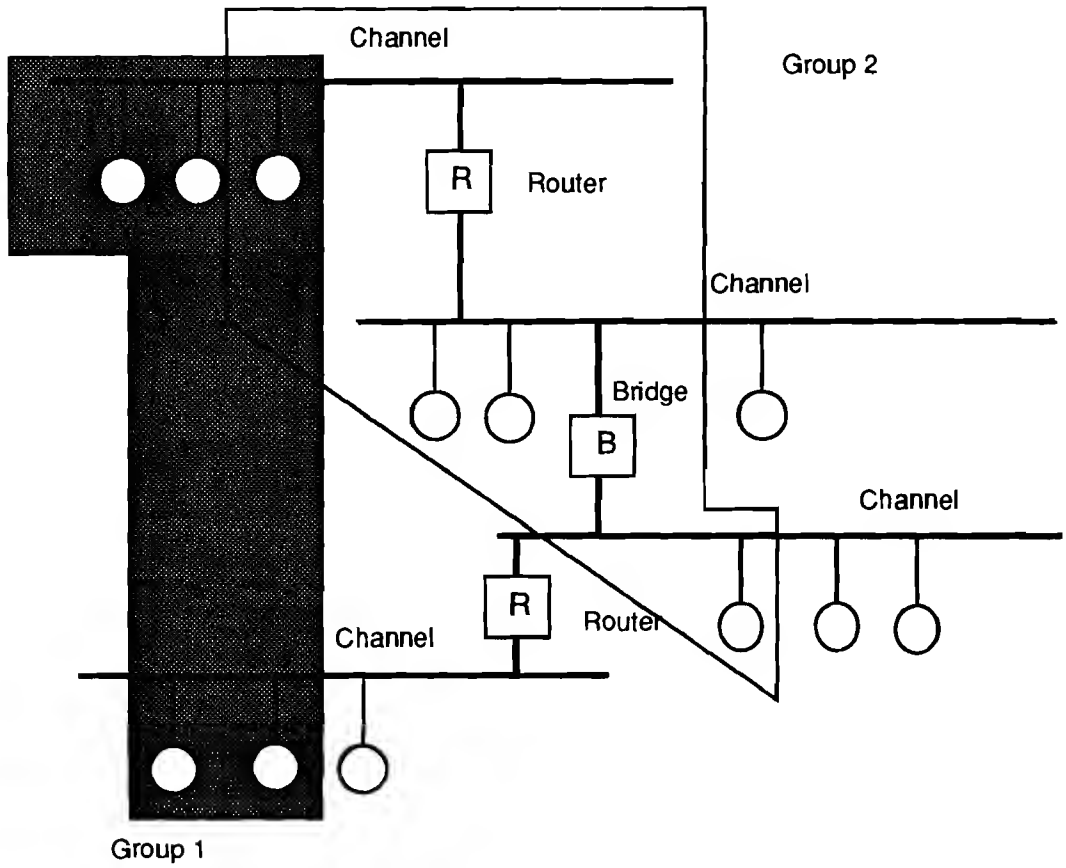


Figure 6. Group membership can span any number of channels, routers, or bridges

Groups are identified by a one byte group number, so a single domain may contain up to 256 groups.

Unique ID

In addition to the Subnet/Node address, a node may always be addressed by its unique ID. The unique ID is 48 bits long, and is assigned when each NEURON CHIP is manufactured. This ID is guaranteed to be unique world-wide.

The *Domain/Unique-ID* addressing format is used by a network management node in the initial configuration of nodes at installation time to assign each node to one or two domains, and to assign subnets and node numbers. There are also other applications for Domain/Unique-ID addressing, such as in networks that perform asset management and inventory control functions.

Addressing Formats

Nodes are addressed using one of five addressing formats. The particular addressing format used determines the number of bytes required for the source and destination address. Table 4 defines the formats and number of bytes required for each. The total address size is computed by adding the appropriate number of bytes indicated in the table to the size of the domain ID, which can range from 0 to 6 bytes depending on the configured size of the domain ID.

Table 4. LONTALK Protocol Address Formats

Address Format	Destination	Address Size (bytes)
Domain (Subnet = 0)	All nodes in the domain	3
Domain, Subnet	All nodes in the subnet	3
Domain, Subnet, Node	Specific node within a subnet	4
Domain, Group	All nodes in the group	3
Domain, Unique-ID	Specific node	9

Network Management and Address Generation

Depending on the level of a given application, a LONWORKS network may or may not require the use of a Network Management node. A Network Management node is a node that has been specifically designated to perform network management functions, such as:

- Find unconfigured nodes and download network addresses
- Stop, start, and reset node applications
- Access node communication statistics
- Configure routers and bridges
- Download new application programs
- Extract the topology of a running network

In a development environment, the role of the network management node is typically performed by the LONBUILDER™ Network Manager. The LONBUILDER Network Manager includes the tools required to define, configure, load, and control LONWORKS networks. The LONBUILDER Protocol Analyzer provides capability to monitor, collect, and display network traffic and performance statistics.

The LONBUILDER software examines the connections between nodes and assigns all groups automatically, in a manner as to optimize network traffic.

Routers and Bridges

A router (or bridge) is a special node that consists of two connected NEURON CHIPS, each connected to a separate channel. Routers and bridges pass packets back and forth between these channels.

There are four types of routers:

- **Repeater**. A repeater is the simplest form of router, simply forwarding all packets between the two channels. Using a repeater, a subnet can exist across multiple channels.
- **Bridge**. A bridge simply forwards all packets which match its domains between the two channels. Using a bridge, a subnet can exist across multiple channels.
- **Configured Router**. Like a learning router, a configured router selectively routes packets between channels by consulting internal routing tables. Unlike a learning router, the contents of the internal routing tables are specified using Network Management commands.
- **Learning Router**. A learning router monitors the network traffic and learns the network topology at the domain/subnet level. The learning router then uses its knowledge to selectively route packets between channels.

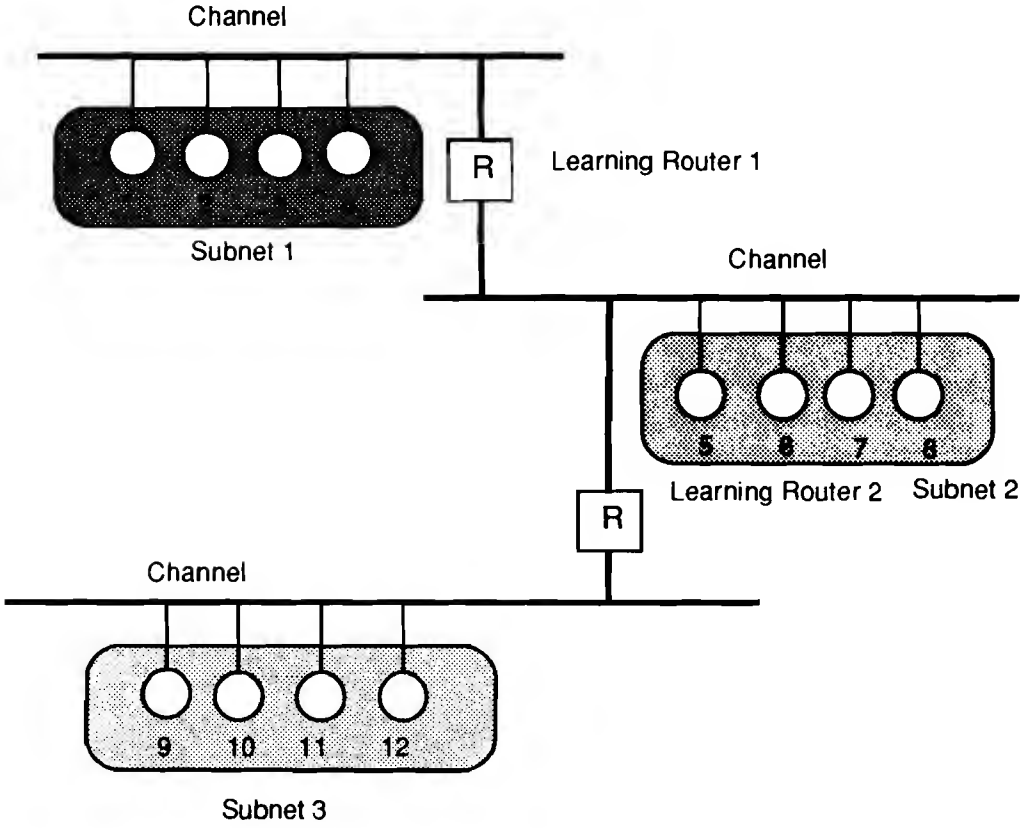


Figure 7. Learning Routers

Initially, each router sets its internal routing tables to indicate that all subnets could lie on either side of the router. Referring to Figure 7, suppose that node 6 generates a message bound for node 2. Learning router 1 initially picks up the message. Examining the source subnet field of the message, the learning router notes in its internal routing tables that subnet 2 lies below it. The router then compares the source and destination subnet IDs - since they are different, the message is passed on.

Meanwhile, learning router 2 has also passed on the message, making an appropriate notation in its internal routing tables regarding the location of subnet 2.

Suppose now that node 2 generates an acknowledgement. This acknowledgement is picked up by learning router 1, who now notes the location of subnet 1. Learning router 1 examines its internal routing tables, and, upon discovering that subnet 2 lies below, passes the message on. When the message appears on subnet 2, it is noted by both node 6 (the destination node), and learning router 2, who does not pass it on but merely notes that subnet 1, like subnet 2, lies somewhere above. Learning router 2 will not learn of the existence or location of subnet 3 until a message is originated from there.

When choosing between learning and configured routers, the following should be taken into account:

- The initial flood of traffic that occurs while a learning router is learning the network topology may cause congestion problems.
- The network topology may have inadvertent "loops" - common in powerline and RF networks - that can cause a learning router to develop an inaccurate network image.
- A learning router is always learning, and will update its internal routing tables to follow changes in network topology.
- The internal routing tables in a learning router do not have to be explicitly programmed.
- Learning routers do not learn about groups but configured routers can be configured to selectively forward group addressed packets.

Subnets cannot cross routers. While bridges and repeaters allow subnets to span multiple channels, the two sides of a router must belong to separate subnets. The fact that routers are selective about the packets that they forward to each channel can be used to increase the total capacity of a system in terms of nodes and connections. In general, it is always a good idea to segment traffic among "communities of interest" if possible.

Communications Services: Efficiency, Response Time, Security, and Reliability

There are a number of tradeoffs relating to network efficiency, response time, security, and reliability: using acknowledged service is most reliable, but uses greater network bandwidth than unacknowledged or unacknowledged repeated service for large groups; prioritizing packets will ensure that those packets will be sent in a timely fashion, to the detriment of others; adding authentication service to designated transactions adds a level of security, but requires twice the number of packets to complete a transaction as non-authenticated transactions.

Selecting Message Services for Reliability and Efficiency

The LONTALK protocol offers four basic types of message service:

The most reliable service is *acknowledged*, or end-to-end acknowledged service, where a message is sent to a node or group of nodes and individual acknowledgements are expected from each receiver. If an acknowledgement is not received from all destinations, the sender times out and re-tries the transaction. The number of re-tries and the time-out are both selectable (see *LONTALK Protocol Timers*, later in this document). The acknowledgements are generated by the network CPU without intervention of the application. Transaction IDs are used to keep track of messages and acknowledgements so that the application does not receive duplicate messages.

An equally reliable service is *request/response*, where a message is sent to a node or group of nodes and individual responses are expected from each receiver. The incoming message is processed by the application on the receiving side before a response is generated. The same retry and time-out options are available as with acknowledged service. Responses may include data, so that this service is particularly suitable for remote procedure call, or client/server applications.

The next most reliable is *unacknowledged* repeated, where a message is sent to a node or group of nodes multiple times, and no response is expected. This service is typically used when broadcasting to large groups of nodes, in which the traffic generated by all the responses would overload the network.

The least reliable is *unacknowledged*, where a message is sent once to a node or group of nodes and no response is expected. This is typically used when the highest performance is required, network bandwidth is limited, and the application is not sensitive to the loss of a message.

Collision Detection

The LONTALK protocol uses a unique collision avoidance algorithm which has the property that under conditions of overload, the channel can still carry close to its maximum capacity, rather than have its throughput degrade due to excess collisions.

When using a communications medium that supports hardware collision detection (twisted pair, for example), the LONTALK protocol can optionally cancel transmission of a packet as soon as a collision is detected by the transceiver. This allows the node to immediately retransmit any packet that has been damaged by a collision. Without collision detection, the node would have to wait the duration of the retry time to notice that no acknowledgement was received — at which time it would retransmit the packet, assuming acknowledged or request/response service. For unacknowledged service, an undetected collision means that the packet is not received and no retry is attempted.

Priority

The LONTALK protocol optionally offers a priority mechanism to improve the response time of critical packets. The protocol permits the user to allocate priority time slots on a channel, dedicated to priority nodes. The network management node that assigns priority slots to individual nodes can ensure that one and only one node is assigned to a particular priority slot on the channel. Each priority time slot on a channel adds a minimum of two bit times to the transmission of every message. The amount of overhead will vary based upon the data rate oscillator accuracy and transceiver requirements. For example, using a LONWORKS 1.25 MBPS twisted pair transceiver with all nodes on the channel having an oscillator accuracy of 0.2% or better, each priority slot is 30 bit times wide. Because there is no contention for the media during the priority portion of a packet cycle, nodes configured with priority have better response time than non-priority nodes. The combination of priority and collision detection allows for bounded response time.

The priority slot assigned to a node applies to all priority packets sent from that node, One, all, or some of the packets sent from a node may be marked as using priority service. The priority designation within a node is made on a per network variable or per message tag basis, and may be set at compile time. In the case of network variables, the priority designation can optionally be changed during or after installation.

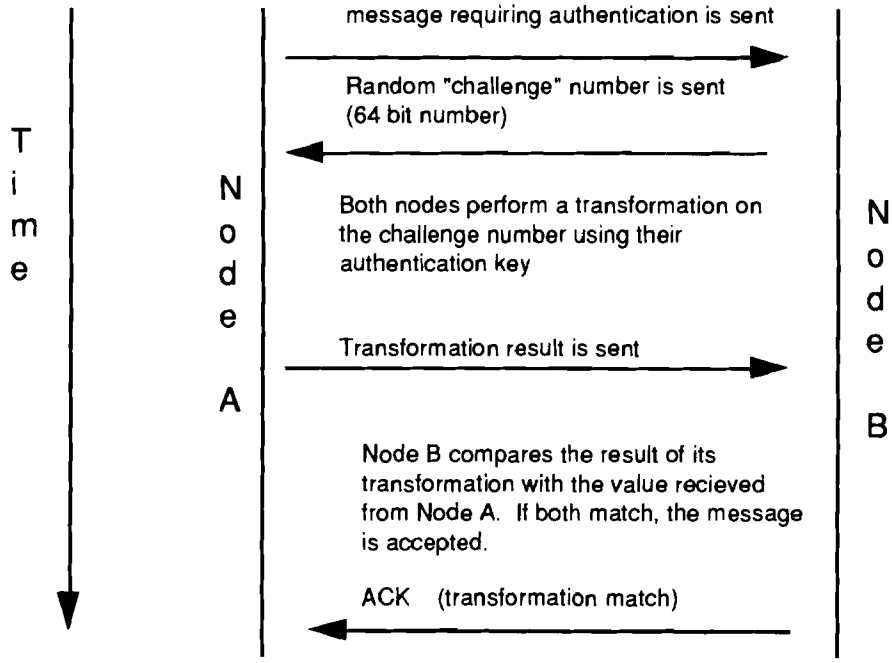
Lower priority numbers indicate higher levels of priority: a priority packet from a node with priority slot #2 will be transmitted before a priority packet from a node with priority slot #4. Setting a node's priority to 0 indicates that none of its packets will be transmitted in a priority slot, regardless of the message service assignment made at node compilation or installation time. Slot #1 is typically reserved for a network management node, to ensure that no application can render a channel incapable of interruption by a Network Manager. Slots 2 through 127 (depending on the medium, and the number of slots allocated on the channel) are then available for prioritized packets from designated nodes.

When a priority packet is generated within a node, it travels out of the node on the priority queue, ahead of any pending non-priority packets buffered for transmission. Similarly, when a priority packet reaches a router or bridge, it goes to the head of the router queue (behind any other queued priority packets) and is forwarded to the far channel using the router's priority slot if one has been configured.

Authentication

The LONTALK protocol supports authenticated messages; the receivers of a message determine if the sender is authorized to send that message. This can prevent unauthorized access to nodes and their applications. The use of authentication is configured individually for each network variable. Network management transactions may also be optionally authenticated.

Authentication is implemented by distributing 48-bit keys to the nodes at installation time, with the sender and receiver of an authenticated message both possessing the same key. When an authenticated message is sent, the receiver challenges the sender to provide authentication, each time using a different random challenge. The sender then uses the authentication key to perform a transformation on the challenge, and responds. The receiver compares the reply to the challenge with its own transformation on the challenge. If the transformations match, the transaction succeeds. The transformation used is designed so that is is extremely difficult to deduce what the key is, even if the challenge and response are both known.



LONTALK Protocol Timers

There are several timers which need to be properly set in order for the LONTALK protocol to function efficiently. These include:

- Transaction Timer
- Repeat Timer
- Group Receive Timer

- Non-group Receive Timer
- Free-buffer Wait Timer

The following section details, for each of the four messaging services, how packets flow through the NEURON CHIP and where the timers come into play.

Packet Flow through the NEURON CHIP

There are four different messaging services in the LONTALK protocol: Unacknowledged, Acknowledged, Unacknowledged/Repeated, and Request/Response. Usually, these services can be used with any addressing mode: domain-wide broadcast, unicast, multicast, or NEURON CHIP ID. There are two exceptions and caveats: first, when performing a broadcast request/response, the application will receive only the first response; all others will be discarded by the network processor. Second, broadcast acknowledged transactions complete once a single acknowledgment is received.

Unacknowledged: When this service is used, the only timer that is involved is the free buffer wait timer. This timer determines the maximum length of time the NEURON CHIP will wait for a free buffer when sending a message. This timer can be deactivated (the NEURON CHIP will wait forever) by setting the timer value to zero. If it is set to another number, n, then the firmware will wait between 2n and 2n + 1 seconds. For example, if the configured number n is set to 2, then the firmware will wait for a free buffer for between 4 and 5 seconds. If a buffer is not obtained before the timer expires, the NEURON CHIP assumes a fatal error and resets.

Acknowledged: Acknowledged service also uses the free buffer wait timer, but additional timers are necessary. Some additional timers are also involved:

1. The **transaction timer** determines how long the node waits for an acknowledgement before retrying. The value of the transaction timer used by the node is taken from the transmitting node's address table entry for the destination address of the packet being sent. The transaction timer is individually configurable by destination address in the address table. If the node does not receive an acknowledgement before the transaction timer expires, it will retry, sending the same packet again (along with an indication of which nodes did acknowledge, in the case of group addressing). This retry process will continue until the **retry count** has been exhausted or until all acknowledgements have been received. The retry count is configurable from 0 to 15 by address table entry.

   Note that a packet may go through routers and bridges to reach its final destinations. The transaction timer should be just long enough so that a packet can reach the "furthest" destination and the acknowledgement from this destination can be received before the transaction timer expires. If the transaction timer is too short, excess retries will be generated; if too long, the time for a transaction to complete will increase on average.

   The LONBUILDER system and the Network Management API automatically calculate the transition time defaults based upon topology, data rate, and the NEURON CHIP input clock frequency.

2. When a packet arrives at its final destinations, the receiving nodes look at the packet's source address and transaction ID. If no "receive transactions" are active with this source address/transaction ID pair, a new receive transaction record is created. If no receive transaction record exists because the node has used them all up with active transactions, the incoming message is lost. Assuming that the receiving node can allocate a receive transaction record, it starts a **receive timer**. It chooses which receive timer to use based upon the address mode that the transmitter used. If the transmitter used group addressing, there exists an address table entry for that group, and the **group receive timer** value is taken from that entry in the address table. If any other addressing mode is used, the node uses its **non-group receive timer** value.

   When the receive timer expires, the transaction record is deleted, and any new transmission having the same transaction ID from the same source address will be treated as a new transaction. Therefore, this timer must be greater than the greatest product of retry count and transaction timer that can be received from the transmitter. A good rule of thumb for setting this timer is *((retry count + 2) * transaction timer)*.

   If the receive transaction timer is too long, then it is likely that the node will run out of memory for receive transaction buffers. If it is too short, then the node may mistake legitimate retries for new transactions, causing duplicate messages to mistakenly be passed on to the application for processing. A good rule of thumb is to keep the retry count low (say, 4) and design networks with as few end-to-end hops as possible; this will keep the transaction timer short. As an example, for a received message that originates from a node with a retry count of 4 and a transaction timer of 200 milliseconds, use the rule-of-thumb above to arrive at ((4 + 2) * 200), or 1200 milliseconds. A shorter value could result in retries being interpreted as new transactions; a longer value could result in a node's running out of receive transaction buffers and losing incoming messages.

Unacknowledged/Repeated: This service follows essentially the same message flow of acknowledged service, with some exceptions. In the address table of the transmitter there is a separate timer known as the **repeat timer**. This timer specifies how frequently the message is repeated when using unacknowledged/repeated service. This time can be shorter than the transaction timer, because no acknowledgement is expected (no time for the acknowledgement need be allotted) when these messages are sent. Transaction IDs and duplicate detection are in effect for these transactions. The transmitter sends the message n times at intervals of m milliseconds, where n is the retry count and m is the repeat timer value.

Request/Response: The message flow for this service is identical to acknowledged service, except that the application sends a response in lieu of an acknowledgement. The fact that the application program adds extra processing time to the generation of the acknowledgement should be taken into account when setting the transaction and receive timers.

Network Management Services

The LONTALK protocol provides network management services for installation and configuration of nodes, downloading of software, and diagnosis of the network. Message types in the network management and diagnostic class are summarized in Table 5.

Table 5. Network Management Messages

Message	From → To	Action	Comments
Query ID*	Net Mgr →Broadcast	report node's unique 48-bit ID	Used to get 48-bit IDs of nodes which are configured or unconfigured, or have a matching node type
Response to Query*	Net Mgr →Node— or — Net Mgr →Broadcast	set node's "response to Query ID" state	used during DB recovery
Update Domain	Net Mgr →Node	assign a node to a domain	propagates encryption key in the clear
Leave Domain	Net Mgr →Node	remove a node from a domain	
Security	Net Mgr →Node	add an increment to the current encryption key to form a new key	does not send encryption key directly, for security
Modify Address Table	Net Mgr →Node	change an address table entry on the node	sets address and timer info. No check performed for duplicate addresses or groups.
Report Address	Net Mgr →Node	report an entry in the address table	response includes address and timer info
Report Net Variable	Net Mgr →Node	report an entry in the network variable configuration table	response includes NVID, priority, direction, service, security
Update Address Data	Net Mgr →Node	update a group entry in the address table	updates group size, timers, retry count
Report Domain	Net Mgr →Node	reports domain info	response includes encryption key
Modify Net Variable	Net Mgr →Node	add or modify entries in the network variable configuration table	sets priority, direction, NVID, service, security
Set Node Mode	Net Mgr →Node	puts the node's application in off-line or on-line state, or resets the node	result should be verified with status request. Typically used to suspend application during EEPROM downloading.

Read Memory	Net Mgr →Node	read any memory location in the node	application code, NV Fixed table, and Event table can be read protected
Write Memory	Net Mgr →Node	write any memory location in the node (subject to write permission)	can be used to download an application into a node. Can restart the NEURON CHIP after writing. Confirm restart with read memory request. A single write should not cross an EEPROM memory boundary.
Checksum Recalculate	Net Mgr →Node	recompute EEPROM checksum	checksum computed over network and application images
Wink	Net Mgr →Node	cause node to execute wink clause in application program	used to identify a node installed on a network but not yet configured
Memory Refresh	Net Mgr →Node	refresh memory	used to extend data retention time of EEPROM
SNVT_Fetch	Net Mgr →Node	retrieve SNVT information	gets SNVT information from from a node where the NEURON CHIP is a communication chip attached to a mircoprocessor containing the node's SNVT information
Network Variable Value Fetch	Net Mgr →Node	retrieve variable information	used to poll network variables by NV index
Status*	Net Mgr →Node	retrieve network error statistics accumulators	response contains: #TX errors, #transaction timeouts, #times no receive transaction memory was available, #lost messages, #missed messages, cause of most recent reset, node state, ROM version, most recent error, and model number.
Clear Status	Net Mgr →Node	clear network error statistics accumulators	clears statistics info, reset cause, error log
Proxy Command*	Net Mgr →Node'Node	request node to deliver command to another node	Net Mgr sends command for node A to send a request to node B. Node B sends response to A, which in tum responds to Net Mgr.
Retrieve Transceiver Status	Net Mgr →Node	retrieve transceiver status registers	response consists of the contents of all 7 status registers in associated transceiver

A system may be configured so that the network management messages listed above (except those marked with an *) are subject to authentication protection independent of application-level messages. This means that only authorized network manager nodes may request these functions.

The Modify Address Table and Modify Net Variable messages may be used to dynamically bind network variables and message tags. This is used during installation and reconfiguration to establish the addressing information needed to route messages and network variable updates between nodes.

Data Interpretation

The LONTALK protocol employs a data oriented application protocol. In this approach, application data items such as temperatures, pressures, states, text strings, and other data items are exchanged between nodes in standard engineering and other predefined units. The command functions are then encapsulated within the application programs of the receiver nodes rather than being sent over the network. In this way, the same engineering value can be sent to multiple nodes which each have a different application for that data item.

Network Variables

The data items in the LONTALK application protocol are called network variables. Network variables can be any single data item or data structure. Application writers need only declare these variables using the keyword 'Network', and the variable will be available to any other node within the network. When "output" network variables change via assignment operations within the application program, the executive built into the NEURON CHIP firmware automatically propagates the new value over the network using LONTALK protocol services. This implicit messaging frees the application writer from buffer management, message initialization, message parsing, and error handling.

Foreign Frame Transmission

A special range of message codes is reserved for foreign frame transmission. Up to 229 bytes of data may be embedded in a message packet and transmitted like any other message. The LONTALK protocol applies no special processing to foreign frames - they are treated as a simple array of bytes. The application program may interpret the data in any way it wishes.

Application Compatibility

Application compatibility is facilitated through the use of Standard Network Variable Types, or SNVTs. The initial list of SNVTs includes nearly 100 types and covers a very wide range of applications. The definition of a SNVT includes units, a range, and a resolution. Using the appropriate network management commands, a LONWORKS node can extract the SNVT information (ID # and optional text string) from any other node. Currently defined SNVTS are listed in the SNVT Guide.

Protocol Services and Parameters

LONTALK protocol services may be selected by a node's application program or by a network management node. In general, a network management node may override services selected by the application program; however, in some instances it may not be overridden.

During development, the LONBUILDER Network Manager performs the role of the network management node that will ultimately be used to install nodes in the final application. Services selected and configured by the LONBUILDER Network Manager at node development (or manufacturing) time may be different than the services selected and configured by the network management node in the final application.

Table 6. Settable Network Image Parameters

Parameter	When/where initialized	Basis for Configuration	Changeable when node is installed?	Compile-time option to prevent field-override of initial setting?
Channel Data Rate	Compilation or installation	Per Node	Yes	No
Domain ID	Installation	Per Domain	Yes	No
Subnet/Node Address	Installation	Per Domain	Yes	No
Group Address(es)	Installation	Per Node	Yes	No
Node Unique ID*	Manufacture	Per Node	No	No
Acknowledged Service - Explicit Messages	Compilation	Per Network Variable or Explicit Message	No	No
Acknowledged Service - Network Variables	Compilation or Installation	Per Network Variable or Explicit Message	Yes	Yes
Retry Count	Installation	Per Network Variable or Explicit Message	Yes	No
Authenticated Service - Explicit Messages	Compilation	Per Network Variable or Explicit Message	No	No
Authenticated Service - Network Variables	Compilation or Installation	Per Network Variable or Explicit Message	Yes	Yes
Parameter	When/where initialized	Basis for Configuration	Changeable when node is installed?	Compile-time option to prevent field-override of initial setting?
Authentication Key	Compilation or Installation	Per Domain	Yes	No
Number of Priority Slots	Installation	Per Node	Yes	No
Priority Service - Explicit Messages	Compilation	Per Network Variable or Explicit Message	No	No
Priority Service - Network Variables	Compilation or Installation	Per Network Variable or Explicit Message	Yes	Yes
Network Variable Types	Compilation	Per Network Variable or Explicit Message	No	No

* Fixed at the time the NEURON CHIP is manufactured. Cannot be modified at any time.

Limits and Bounds

LONTALK Domain IDs may be 0, 1, 3 , or 6 bytes in length. All nodes in a common domain must have identical Domain IDs of the same length. Within each LONTALK Domain, the following limits apply:

LONTALK Limits

- A maximum of 225 subnets
- A maximum of 127 nodes per subnet
- A maximum of 256 groups
- A maximum of 64 nodes per group (acknowledged services only -- there is no node limit on groups using unacknowledged services)
- A maximum of 32,385 nodes
- A node has one Subnet and one Node address per Domain to which it belongs
- Group membership must be in a Domain to which the node belongs
- A node may have up to 255 network variables defined.

LONWORKS Node Limits

- A node based on a NEURON 3120 or NEURON 3150 CHIP may belong to a maximum of 2 Domains
- A node may have a single outgoing transaction in progress at a time
- A node may have a singe authenticated transaction in progress at a time
- An application based on a NEURON 3120 or NEURON 3150 CHIP may have up to 62 network variables defined
- A node based on a NEURON 3120 or NEURON 3150 CHIP may be a member of up to 15 groups
- All nodes must be able to receive a 60-byte layer 2 frame
- All nodes must be able to send a 32-byte layer 2 frame

NEURON CHIP Network Image Road Map

The following items are among the contents of the EEPROM in every NEURON CHIP:

Unique ID	6 bytes, set at chip manufacture time, cannot be changed. This is the address used in installation and network management. This ID is unique worldwide.
Mfg Data	2 bytes, set at chip manufacture time, cannot be changed. Identifies model number; e.g., Motorola 3120.
Node Type	8 bytes, set at application compile time. In non-certified nodes, contains the NEURON C program name. In certified nodes, this field contains the manufacturer's ID, the node type and sub type, etc. See the NEURON C Programmer's Guide for details of the bit fields.
Node Address	One or two structures, each containing a domain (0, 8, 24, or 48 bits long), an authentication key (48 bits), a node number (8 bits), and a subnet number (8 bits). There are two of these if the node can be a member of two domains.
Address Table	(see the following section)
Location ID	2 byte channel number, 6 byte location string. Set at installation time to correspond to a physical location represented on a building plan, etc.
NV Config	Table with an entry for each network variable. For each network variable: priority bit, direction (input or output), network variable ID (assigned by binder), protocol service to use, authenticated bit, and address table index.
NV fixed	Table with an entry for each network variable describing the compile time attributes: synchronized, SI-SNVT/SD Data index, NV length in bytes, NV address.
Comm Data	The number of priority slots on this channel, this node's priority slot, data rate, and transceiver parameters.
Mode Table	Node's configuration; e.g., how many domains, how many address table entries, how many service interface control blocks (sicb's), how many MAC layer buffers, number of receive transaction structures, number of address table entries, number of network variables.
SI-SNVT/SD	(Self- identifying Standard Network Variable Type with Data optional Self-Documentation) SNVT index if SNVT's are used, and optional self documentation data about the node. Variable length structure with optional text or other data for each NV which uses a SNVT.

The Address Table

The size of the address table of a node is configurable by the application writer. The default specifies 15 separate destination addresses in the address table. The address table is used to get the address to send a message, and, in the case of group addressing, is used to determine if the message received is from a group where the node is a member. For each address table entry, the first byte contains the type of address table entry that follows. There are three types defined: group, subnet/node, and broadcast.

The group structure has entries for the group size (how many acknowledgements to expect), which domain to use, what this node's group member number is, (to identify an acknowledgement as coming from this node), a transmit timer, a repeat timer, a retry count, a receive timer, and the group ID.

The subnet/node structure has entries for the domain, destination node number, destination subnet number, a repeat timer, a retry count, and a receive timer.

The broadcast structure has entries for the domain and destination subnet (0 implies all subnets), a repeat timer, a retry count, and a receive timer.

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304
Telephone (415) 855-7400
Fax (415) 856-6153

Echelon Europe Ltd
105 Heath Street
London NW3 6SS
England
Telephone (071) 431-1600
Fax (071) 794-0532
International Telephone + 44 71 431-1600
International Fax + 44 71 794-0532

Echelon Japan K.K.
AIOS Gotanda Building #808
10-7, Higashi-Gotanda 1-chome,
Shinagawa-ku, Tokyo 141, Japan
Telephone (03) 3440-8638
Fax (03) 3440-8639

Part Number 005-0017-01